

What is claimed is:

1. A computer program embodied on an electronic medium, wherein the computer program is configured to:

5 create a scene graph by:

adding one or more objects to a hierarchy by defining parent-child relationships for the one or more objects;

generate a parallel representation of the scene graph by:

10 determining whether a data structure corresponding to the one or more object types already exist;

if a corresponding data structure does not already exist, create a new data structure corresponding to the type of object;

if a corresponding data structure does already exist, create a new entry in the corresponding data structure;

15 create at least one thread configured to process each data structure; and

render the scene graph by executing at least a subset of the threads.

2. The computer program of claim 1, wherein the data structures provide a parallel context from which to access the objects in the scene graph.

20 3. The computer program of claim 1, wherein the data structures provide alternate access to data from the objects in the scene graph.

25 4. The computer program of claim 1, wherein the data structures provide an alternate ordering of information in the scene graph.

5. The computer program of claim 1, wherein the entries in the data structures comprise pointers to the data in the scene graph.

30 6. The computer program of claim 1, wherein at least a subset of the entries in the data structures are copies of the data in the scene graph.

7. The computer program of claim 1, wherein the objects in the scene graph comprise object data,
wherein the entries in the data structures comprise pointers to the object data.

8. The computer program of claim 1, wherein the entries in the data structures comprise pointers to
objects in the scene graph.

9. The computer program of claim 1, wherein said adding one or more objects to a hierarchy by
defining parent-child relationships for the one or more objects comprises inserting a pointer to
an object in the hierarchy.

10. The computer program of claim 1, further comprising generating one or more threads for each
data structure, wherein the threads operate on the data structures.

11. The computer program of claim 1, further comprising generating one or more threads for each
data structure, wherein the threads operate on the data structures.

12. A method for managing a virtual world, the method comprising:

creating a scene graph;

adding one or more pointers to the scene graph, wherein the pointers point to objects that are part of
the virtual world;

generating a parallel means of accessing the one or more objects in the scene graph by:

determining whether a data structure corresponding to the one or more objects already exist,

and,

if the corresponding data structure does not already exist, creating a new data structure

corresponding to the type of object;

creating a new entry in the corresponding data structure;

creating one or more threads to act on the data structures; and

executing the threads to render an image corresponding to the scene graph.

13. The method of claim 12, wherein the objects include one or more of the following:

three-dimensional graphical objects;

lights;
transformations; and
environmental attributes.

- 5 14. The method of claim 12, wherein the objects include sound data files.
15. The method of claim 12, wherein the objects include haptic data files.
16. The method of claim 12, wherein the data structures provide a parallel context from which to
10 access the objects in the scene graph.
17. The method of claim 12, wherein the data structures provide alternate access to data from the
objects in the scene graph.
- 15 18. The method of claim 12, wherein the data structures provide an alternate ordering of
information in the scene graph.
19. The method of claim 12, wherein the entries in the data structures comprise pointers to the data
in the scene graph.
- 20 20. The method of claim 12, wherein at least a subset of the entries in the data structures are copies
of the data in the scene graph.
21. The method of claim 12, wherein the objects are added to the scene graph by creating pointers
25 to data files in a memory.
22. The method of claim 12, wherein the entries in the data structures comprise pointers to the
objects in the scene graph.
- 30 23. The method of claim 12, wherein said adding one or more objects to a hierarchy by defining
parent-child relationships for the one or more objects comprises inserting a pointer to a data file
corresponding to the object into the hierarchy.

24. The method of claim 12, further comprising generating one or more threads for each data structure, wherein the threads operate on the data structures.

5 25. The method of claim 12, further comprising generating one or more threads for each data structure, wherein the threads operate in parallel on the data structures.

26. The method of claim 12, further comprising transmitting one or more of the data structures to a remote processor for remote processing.

10 27. The method of claim 12, further comprising transmitting one or more of the threads to a remote processor for remote processing.

15 28. A computer program embodied on an electronic medium, wherein the computer program is configured to:

create a scene graph by:

adding a first object to a hierarchy;

adding a second object to the hierarchy;

defining a parent-child relationship between the first and second objects;

20 creating a plurality of data structures and threads, wherein each data structure corresponds to a particular type of object in the hierarchy, wherein each object in the hierarchy has an entry in at least one of the data structures, and wherein each thread is configured to execute one of the data structures; and

rendering the scene graph by executing the threads in lieu of traversing the scene graph.

25 29. The computer program of claim 28, wherein each data structure has a corresponding update thread configured to update the data structure.

30 30. The computer program of claim 28, wherein the scene graph includes behavior data for the objects.

31. The computer program of claim 28, wherein the scene graph includes environmental data.

32. The computer program of claim 28, wherein the scene graph includes lighting and fog information.

5 33. The computer program of claim 28, wherein the program is configured to optimize the scene graph before creating the plurality of data structures and the threads.

34. The computer program of claim 28, wherein the program is configured to optimize the data structures by stripifying geometry data in the data structure.

10 35. The computer program of claim 28, wherein the program is configured to optimize the data structures by splitting nodes in the data structures.

15 36. The computer program of claim 28, wherein the program is configured to optimize the data structures by flattening the scene graph before creating the plurality of data structures and the threads.